

REMARKS

The examiner rejected claims 1, 8, 14 and 27 under 35 U.S.C. §102(e) as being anticipated by US PG publication No. 2003/0217223 to Nino.

The examiner also rejected claims 2-4, 6, 13, 15-18, 25-26 and 28-30 under 35 U.S.C. §103(a) as being unpatentable over Nino in view of U.S. Patent No. 6,571,319 to Tremblay et al.

The examiner also rejected claims 7 and 9-12 under 35 U.S.C. 103(a) as being unpatentable over Nino in view of the Microsoft Press Computer Dictionary, and rejected claim 19 and 21-24 under 35 U.S.C. 103(a) as being unpatentable over Nino and Tremblay, in view of the Microsoft Press Computer Dictionary

Specifically, with respect to applicant's claim 1, the examiner stated:

3. With respect to claim 1, Nino teaches a method comprising:
 - converting memory access instructions into standard formatted memory access instructions, in pars 21-23 where the memory access read or write is divided into the standard format memory access commands, precharge, activate, and read (or write).
 - generating a plurality of memory access partitions containing corresponding subsets of the intermediary standard formatted memory access instructions, with the plurality of memory access partitions directed to specific memory banks. In pars. 21-23 where a plurality of memory access partitions consists of a precharge an activate and a read or write directed towards memory bank A or B.
 - generating a match set of instruction patterns. Including matches of instruction patterns to the corresponding subsets of the intermediary standard formatted memory access instructions in the plurality of memory access instructions. In pars. 26-27 where the match set is the two instructions that are combined; and
 - transforming the matches to vector memory access instructions, in pars. 26-27. (Final Office Action, pages 2-3)

Applicant disagrees.

Applicant's claim 1 recites "converting memory access instructions in a source code into intermediary standard formatted memory access instructions; generating a plurality of memory access partitions containing corresponding subsets of the intermediary standard formatted memory access instructions, with the plurality of memory access partitions directed to specific memory banks; generating a match set of instruction patterns including matches of instruction patterns to the corresponding subsets of the intermediary standard formatted memory access

instructions in the plurality of memory access partitions; and transforming the matches to vector memory access instructions."

Thus, applicant's method, as recited in claim 1, includes converting source-code memory access instructions into intermediary instructions, and also includes transforming matches (of instructions patterns to subsets of the intermediary standard formatted memory access instructions) to vector memory access instructions.

Nino, on the other hand, describes a circuit and method of operation for combining commands in a DRAM (Abstract). Specifically, Nino explains:

[0026] FIG. 4 depicts a combined command embodiment in which each bank in the memory array is read from one time and written to one time. These are the same operations that were performed in FIG. 3, and thus the advantages of the combined commands may be seen in the fewer clock cycles taken to complete the operations, that is, 27 clock cycles in FIG. 4 rather than 31 clock cycles in FIG. 3. The same latency and operational periods described above for FIG. 3 apply to FIG. 4. FIG. 4 is arranged in a manner similar to FIG. 3, with commands to each bank under the column headings, A, B, C and D. There are now two columns labeled "COMMANDS" because more than one command may be given at once. Input/output to the DRAM is noted under the "I/O" column. Commands to more than one row at a time are called row/row commands and commands to a row and a column at the same time are called row/column commands.

[0027] In this example, commands are combined, as seen in command sequences 20, 22, 24, 25, 26, 27 and 28. Idle time is again depicted by blank boxes. In sequence 20, a combined row command is given to two different banks, activate A and precharge B. The command will be given to the same or different rows in both A and B. In sequence 22, a column command to one bank is combined with a row command to another bank, Read A and Activate B. In the next sequence 24 a combined command is given to Activate C and Precharge D, that is, to activate a particular row in bank C and precharge that same row or a different row in bank D. Note that the sequence used for reading or writing is not changed from "precharge," "activate," and then "read" or "write." Time is saved by combining commands as shown. If more read and write operations were in progress in FIG. 4, what appears as primarily idle time (blank boxes) would have more combined operations and more time would be saved. As it is in this sequence, the four read and write operations consume 27 command clock cycles, or about 216 ns at 125 MHz (8 ns per command cycle). This saves about 32 ns, about a 15% speed-up of this particular read/write operation for the DRAM of FIG. 2. Other data input/output operations may save more or less time depending on the actual operations needed and taken.

[0028] In order to implement a combined command DRAM, certain modifications should be made to the control logic used for operating DRAMs. Until now, commands were typically issued one-at-a-time, rather than combining commands, with the exception of unique situations such as

an "auto-precharge" or "precharge all," commanding rows only to more than one bank, or write with auto-precharge, combining row and column commands on the same bank. By contrast, embodiments of the present invention combine commands either to rows in multiple banks, or to rows and columns in multiple banks. (Emphasis added, FIGS. 3 and 4 and pages 2-3, paragraphs 26-28)

Nino's commands are combinations of two or more DRAM commands that are executed simultaneously rather than at separate times, as has been done by the prior art that Nino alludes to. At no point does Nino describe that commands are converted into intermediary commands having a standard format.

Furthermore, even assuming, for argument sake, that the combined commands could be viewed as intermediary standard formatted commands, Nino does not describe that intermediary commands are matched to command (or instruction) patterns. Indeed, all Nino does is combine discrete commands so that those same commands would be performed simultaneously, thus shortening the execution time that otherwise would be required. Such combining of commands does not involve any type of pattern matching because there is no need to determine the exact nature of the command to be performed.

Further still, at no point does Nino describe that matches of command patterns (or instruction patterns) to subsets of intermediary standard formatted commands are transformed into vector memory access commands (or instructions). As explained, Nino combines discrete commands into combined sequences of commands, and then executes the combined sequences of commands so that discrete commands in a sequence are executed at the same time. Thus, even if the sequences of combined commands can be viewed as an intermediary standard formatted commands (which applicant contends they cannot), Nino does not describe that those sequences are transformed into vector commands.

Accordingly, Nino fails to disclose or suggest at least the features of "converting memory access instructions in a source code into intermediary standard formatted memory access instructions," "generating a match set of instruction patterns including matches of instruction patterns to the corresponding subsets of the intermediary standard formatted memory access instructions in the plurality of memory access partitions" and/or "transforming the matches to vector memory access instructions," as required by applicant's independent claim 1.

Independent claim 1, and the claims that depend from it, are therefore patentable over the cited art.

Independent claims 15 and 27 recite “converting source code that includes memory access instructions that read or write less than a minimum data access unit (MDAU) to intermediary code that includes memory access instructions that read or write a multiple of the minimum data access unit; converting the memory access instructions of the intermediary code into intermediary memory access instructions that have a format including a base address plus an offset; grouping subsets of the intermediary memory access instructions into a plurality of memory access partitions, with the plurality of memory access partitions containing intermediate memory access instructions directed to specific memory banks; and vectorizing the intermediary memory access instructions in the subsets corresponding to the plurality of memory access partitions that match instruction patterns,” or similar language.

For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by the cited art. Applicant's independent claims 15 and 27, and the claims that depend from them, are therefore patentable over the cited art.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Applicant : Bo Huang et al.
Serial No. : 10/718,283
Filed : November 19, 2003
Page : 12 of 12

Attorney's Docket No.: 10559-886001 / Intel Corporation P17581

No fee is believed due. Please apply any required fees to deposit account 06-1050,
referencing the attorney docket number shown above.

Respectfully submitted,

Date:

May 8, 2007

Ido Rabinovitch
Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21633359.doc